```
RRRRRRRRRRR      MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRRRRRRRRRR      MMM          MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSS
RRR   RRR        MMM          MMM             SSS
RRR   RRR        MMM          MMM             SSS
RRR   RRR        MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
```

FILEID**NTOPUT

```
NN      NN  TTTTTTTTTT    OOOOOO    PPPPPPPP    UU      UU  TTTTTTTTTT
NN      NN  TTTTTTTTTT    OOOOOO    PPPPPPPP    UU      UU  TTTTTTTTTT
NN      NN      TT      OO      OO  PP      PP  UU      UU      TT
NN      NN      TT      OO      OO  PP      PP  UU      UU      TT
NNNN    NN      TT      OO    OOOO  PP      PP  UU      UU      TT
NNNN    NN      TT      OO    OOOO  PP      PP  UU      UU      TT
NN  NN  NN      TT      OO  OO  OO  PPPPPPPP    UU      UU      TT
NN  NN  NN      TT      OO  OO  OO  PPPPPPPP    UU      UU      TT
NN    NNNN      TT      OOOO    OO  PP          UU      UU      TT
NN    NNNN      TT      OOOO    OO  PP          UU      UU      TT
NN      NN      TT      OO      OO  PP          UU      UU      TT          ....
NN      NN      TT      OO      OO  PP          UU      UU      TT          ....
NN      NN      TT        OOOOOO    PP          UUUUUUUUUU      TT          ....
NN      NN      TT        OOOOOO    PP          UUUUUUUUUU      TT          ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

(2)	64	DECLARATIONS
(3)	126	NT$PUT - PERFORM NETWORK PUT FUNCTION
(4)	411	NT$UPDATE - PERFORM NETWORK UPDATE FUNCTION
(5)	495	NT$DELETE - PERFORM NETWORK DELETE FUNCTION

```
0000     1          $BEGIN  NTOPUT,000,NF$NETWORK,<NETWORK PUT/UPDATE/DELETE RECORD>
0000     2
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; Facility: RMS
0000    31 ;
0000    32 ; Abstract:
0000    33 ;
0000    34 ;       This module communicates with the file access listener at the remote
0000    35 ;       node to perform the put, udpate, and delete functions.
0000    36 ;
0000    37 ; Environment: VAX/VMS, executive mode
0000    38 ;
0000    39 ; Author: James A. Krycka,      Creation Date:  09-DEC-1977
0000    40 ;
0000    41 ; Modified By:
0000    42 ;
0000    43 ;       V03-005 RAS0315         Ron Schaefer    22-Jun-1984
0000    44 ;               Eliminate tiny data psect.
0000    45 ;
0000    46 ;       V03-004 JAK0145         J A Krycka      12-APR-1984
0000    47 ;               Track changes in DAP message building algorithm.
0000    48 ;
0000    49 ;       V03-003 JAK0126         J A Krycka      08-SEP-1983
0000    50 ;               Return relative record number in RAB$L_BKT field on sequential
0000    51 ;               $PUT operations to a relative file (per RMS documentation).
0000    52 ;
0000    53 ;       V03-002 JAK0122         J A Krycka      22-AUG-1983
0000    54 ;               On $PUT failure, verify that FAL has sent a Status message
0000    55 ;               before sending a Continue Transfer message to unlock FAL.
0000    56 ;               Move NT$RESUME_FAL routine to NTODISCON module and perform
0000    57 ;               other miscellaneous cleanup edits.
```

```
0000     58 ;
0000     59 ;       V03-001 KRM0092         Karl Malik      18-Mar-1983
0000     60 ;               Add support for STMLF and STMCR formats.
0000     61 ;
0000     62 ;--
```

```
                                        .SBTTL  DECLARATIONS

;
; Include Files:
;

                                $BDBDEF                 ; Define BDB symbols
                                $DAPPLGDEF              ; Define DAP prologue symbols
                                $DAPHDRDEF              ; Define DAP message header
                                $DAPCNFDEF              ; Define DAP Configuration message
                                $DAPCTLDEF              ; Define DAP Control message
                                $DAPDATDEF              ; Define DAP Data message
                                $FABDEF                 ; Define File Access Block symbols
                                $IFBDEF                 ; Define IFAB symbols
                                $IRBDEF                 ; Define IRAB symbols
                                $NWADEF                 ; Define Network Work Area symbols
                                $RABDEF                 ; Define Record Access Block symbols
                                $RMSDEF                 ; Define RMS completion codes

;
; Macros:
;
;       None
;
; Equated Symbols:
;

0000000A                LF=10                           ; Line feed
0000000B                VT=11                           ; Vertical tab
0000000C                FF=12                           ; Form feed
00000010                DLE=16                          ; Data link error  (Control-P)
00000011                DC1=17                          ; Device control 1 (Control-Q)
00000012                DC2=18                          ; Device control 2 (Control-R)
00000013                DC3=19                          ; Device control 3 (Control-S)
00000014                DC4=20                          ; Device control 4 (Control-T)
0000001A                CTRLZ=26                        ; Control-Z
0000001B                ESC=27                          ; Escape

00000A0D                CRLF=^X0A0D                     ; ASCII codes for CR and LF
0000000A                LF=^X0A                         ; ASCII code for LF
0000000D                CR=^X0D                         ; ASCII code for CR

                        ASSUME  DAP$Q_DCODE_FLG EQ 0
                        ASSUME  NWA$Q_FLG EQ 0

;
; Own Storage:
;


;
; The following bitmask represents the DAP terminator set for stream (STM)
; format files where each bit position corresponds to an ASCII character code
; for a single-character terminator. Note that the default terminator, CRLF,
; is not expressed in this mask.
;
```

```
                0000   121 NT$STM_MASK::                                   ; DAP stream terminator bitmask
0C1F1C00        0000   122          .LONG   <<1@LF> + <1@FF> + <1@VT> + <1@CTRLZ> + <1@ESC> + -
                0004   123                  <1@DLE> + <1@DC1> + <1@DC2> + <1@DC3> + <1@DC4>>
                0004   124
```

```
0004   126          .SBTTL  NT$PUT - PERFORM NETWORK PUT FUNCTION
0004   127
0004   128 ;++
0004   129 ; NT$PUT - engages in a DAP dialogue with the remote FAL to put (store) the
0004   130 ;       specified record of a sequential, relative, or indexed file. This
0004   131 ;       routine supports both DAP file transfer and record transfer modes.
0004   132 ;
0004   133 ; Calling Sequence:
0004   134 ;
0004   135 ;       BSBW    NT$PUT
0004   136 ;
0004   137 ; Input Parameters:
0004   138 ;
0004   139 ;       R4      BDB address
0004   140 ;       R8      RAB address
0004   141 ;       R9      IRAB address
0004   142 ;       R10     IFAB address
0004   143 ;       R11     Impure Area address
0004   144 ;
0004   145 ; Implicit Inputs:
0004   146 ;
0004   147 ;       BDB buffer contents
0004   148 ;       BDB$L_ADDR
0004   149 ;       BDB$W_NUMB
0004   150 ;       DAP$V_DAPCRC
0004   151 ;       DAP$V_GEQ_V56
0004   152 ;       DAP$L_CRC_RSLT
0004   153 ;       IFB$L_NWA_PTR
0004   154 ;       IFB$V_SQO
0004   155 ;       IRB$B_MODE
0004   156 ;       NWA$V_FTM_INIT
0004   157 ;       NWA$V_FTM_RETRV
0004   158 ;       NWA$W_DAPBUFSIZ
0004   159 ;       NWA$B_RFM
0004   160 ;       NWA$B_ORG
0004   161 ;       NWA$Q_BLD
0004   162 ;       RAB$B_KSZ
0004   163 ;       RAB$B_RAC
0004   164 ;       RAB$L_ROP
0004   165 ;       RAB$L_KBF
0004   166 ;
0004   167 ; Output Parameters:
0004   168 ;
0004   169 ;       R0      Status code (RMS)
0004   170 ;       R1-R3   Destroyed
0004   171 ;       AP      Destroyed
0004   172 ;
0004   173 ; Implicit Outputs:
0004   174 ;
0004   175 ;       BDB$B_FLGS
0004   176 ;       DAP$L_CRC_RSLT
0004   177 ;       IRB$L_IOS and IRB$L_IOS4 zeroed
0004   178 ;       NWA$V_FTM_INIT cleared
0004   179 ;       NWA$V_FTM_STORE
0004   180 ;       RAB$L_BKT
0004   181 ;       RAB$W_RFA
0004   182 ;
```

```
                     0004   183 ; Completion Codes:
                     0004   184 ;
                     0004   185 ;       Standard RMS completion codes
                     0004   186 ;
                     0004   187 ; Side Effects:
                     0004   188 ;
                     0004   189 ;       None
                     0004   190 ;
                     0004   191 ;--
                     0004   192
                     0004   193 NT$PUT::                                    ; Entry point
                     0004   194         $TSTPT  NTPUT                      ;
    00F0 8F    BB    000A   195         PUSHR   #^M<R4,R5,R6,R7>           ; Save registers
    56   54    D0    000E   196         MOVL    R4,R6                      ; Copy BDB address
57   3C AA    D0    0011   197         MOVL    IFB$L_NWA_PTR(R10),R7      ; Get address of NWA (and DAP)
      FFE8'   30    0015   198         BSBW    NT$CHK_RAC                 ; Validate record access mode
     10 50    E9    0018   199         BLBC    R0,10$                     ; Branch on failure
0F 67    1A    E0    001B   200         BBS     #NWA$V_FTM_RETRV,(R7),20$;$PUT after $GET not allowed in FTM
                     001F   201
                     001F   202         ASSUME  RAB$C_SEQ EQ 0
                     001F   203         ASSUME  RAB$C_KEY EQ 1
                     001F   204         ASSUME  RAB$C_RFA EQ 2
                     001F   205
                     001F   206         $CASEB  SELECTOR=RAB$B_RAC(R8)- ; Dispatch on access mode:
                     001F   207                 DISPL=<-                   ;
                     001F   208                     PUT_SEQ-               ;  Sequential record access
                     001F   209                     PUT_KEY-               ;  Access by key value
                     001F   210                 >                          ;  Access by record file address
          014E   31  0028   211         BRW     ERRRAC                     ; Branch aid
          0157   31  002B   212 10$:    BRW     EXIT                       ; Branch aid
          014F   31  002E   213 20$:    BRW     ERRFTM                     ; Branch aid
                     0031   214
                     0031   215 ;+
                     0031   216 ; Sequential access has been requested.
                     0031   217 ;
                     0031   218 ; Note: In record transfer mode a Control message must be sent for each record
                     0031   219 ;       retrieved, whereas, in file transfer mode only an initial Control
                     0031   220 ;       message is required.
                     0031   221 ;-
                     0031   222
                     0031   223 PUT_SEQ:                                   ;
05 6A    2D    E0    0031   224         BBS     #IFB$V_SQO,(R10),10$       ; Branch if in file transfer mode
   51    00    9A    0035   225         MOVZBL  #DAP$K_SEQ_ACC,R1          ; Set RAC for DAP message
         0B    11    0038   226         BRB     PUT_SEQ_COMMON             ; Join common code
   67    19    E5    003A   227 10$:    BBCC    #NWA$V_FTM_INIT,(R7),-     ; Branch if no Control message required
         25          003D   228                 PUT_SEND_DAT               ;  and turn off single-shot flag
                     003E   229         $SETBIT #NWA$V_FTM_STORE,(R7)      ; Set file transfer mode storage flag
   51    03    9A    0042   230         MOVZBL  #DAP$K_SEQ_FILE,R1         ; Set RAC for DAP message
                     0045   231
                     0045   232 PUT_SEQ_COMMON:                           ; Common code for put operation
   50    04    D0    0045   233         MOVL    #DAP$K_CTL_MSG,R0          ; Get message type value
      FFB5'   30    0048   234         BSBW    NT$BUILD_HEAD              ; Construct message header
   85    04    90    004B   235         MOVB    #DAP$K_PUT_WRITE,(R5)+     ; Store CTLFUNC field
   85    09    90    004E   236         MOVB    #<<DAP$M_RAC>!-            ; Store CTLMENU field
                     0051   237                   <DAP$M_ROP>!-            ;
                     0051   238                   0>,(R5)+                 ;
   85    51    90    0051   239         MOVB    R1,(R5)+                   ; Store RAC field
```

```
                     0054   240
                     0054   241 ;+
                     0054   242 ; Common code to build rest of the Control message and send it to partner.
                     0054   243 ;-
                     0054   244
                     0054   245 PUT_SEND_CTL_COMMON:                        ; Used for both put and update
          FFA9'  30  0054   246         BSBW    NT$MAP_ROP                 ; Store ROP as an extensible field
          FFA6'  30  0057   247         BSBW    NT$BUILD_TAIL              ; Finish building message
          FFA3'  30  005A   248         BSBW    NT$TRANSMIT                ; Send Control message to FAL
       03 50     E8  005D   249         BLBS    R0,PUT_SEND_DAT            ; Branch on success
        0122     31  0060   250         BRW     EXIT                       ; Branch aid
                     0063   251
                     0063   252 ;+
                     0063   253 ; Build and send DAP Data message to partner containing user record.
                     0063   254 ;-
                     0063   255
                     0063   256 PUT_SEND_DAT:                             ;
    04 6A    2D     E0  0063   257         BBS     #IFB$V_SQO,(R10),10$       ; Branch if in file transfer mode
                     0067   258         $SETBIT #NWA$V_LAST_MSG,(R7)       ; Declare this last message to block
        50    08     D0  006B   259 10$:    MOVL    #DAP$K_DAT_MSG,R0          ; Get message type value
           FF8F'  30  006E   260         BSBW    NT$BUILD_HEAD              ; Construct message header
54    00F4 C7     D0  0071   261         MOVL    NWA$Q_BLD+4(R7),R4         ; Get address of build message buffer
             85    94  0076   262         CLRB    (R5)+                      ; Zero RECNUM field
        53    55     D0  0078   263         MOVL    R5,R3                      ; Save next byte pointer
  52      14  A6     3C  007B   264         MOVZWL  BDB$W_NUMB(R6),R2          ; Get record size
51    55      54     C3  007F   265         SUBL3   R4,R5,R1                   ; Compute # DAP overhead bytes in msg
55    51      52     C1  0083   266         ADDL3   R2,R1,R5                   ; Compute projected size of DAP message
00CA C7     55     B1  0087   267         CMPW    R5,NWA$W_DAPBUFSIZ(R7)     ; Make sure message will fit in buffer
             38    1A  008C   268         BGTRU   40$                        ; Branch if record is too big
      0120 C7     52  7D  008E   269         MOVQ    R2,NWA$Q_SAVE_DESC(R7)     ; Save descriptor of user record
             20    BB  0093   270         PUSHR   #^M<R5>                    ; Save register
63    18 B6     52     28  0095   271         MOVC3   R2,@BDB$L_ADDR(R6),(R3)    ; Move record into DAP message
             20    BA  009A   272         POPR    #^M<R5>                    ; Restore register
                     009C   273
                     009C   274         ASSUME  FAB$C_UDF EQ 0
                     009C   275         ASSUME  FAB$C_FIX EQ 1
                     009C   276         ASSUME  FAB$C_VAR EQ 2
                     009C   277         ASSUME  FAB$C_VFC EQ 3
                     009C   278         ASSUME  FAB$C_STM EQ 4
                     009C   279         ASSUME  FAB$C_STMLF EQ 5
                     009C   280         ASSUME  FAB$C_STMCR EQ 6
                     009C   281
                     009C   282         $CASEB  SELECTOR=NWA$B_RFM(R7)-    ; Dispatch on record format:
                     009C   283                 BASE=#FAB$C_STM-           ;
                     009C   284                 DISPL=<-                   ;
                     009C   285                     20$-                   ;  STM
                     009C   286                     60$-                   ;  STMLF
                     009C   287                     70$-                   ;  STMCR
                     009C   288                 >                          ;  UDF, FIX, VAR, VFC
        50    11     00A8   289         BRB     100$                       ; Bypass stream specific processing
                     00AA   290
                     00AA   291 ;
                     00AA   292 ; The remote file is in stream format--add a CRLF to the user record if it
                     00AA   293 ; does not end in a stream form effector character.
                     00AA   294 ;
                     00AA   295
          14 A6      B5  00AA   296 20$:    TSTW    BDB$W_NUMB(R6)             ; Branch if this is a null record
```

```
                0F   13   00AD   297          BEQL    30$                             ;
          50    FF A3   9A   00AF   298          MOVZBL  -1(R3),R0                       ; Get record terminator character
              1F   50   91   00B3   299          CMPB    R0,#31                          ; Branch if out-of-range for a stream
                   06   1A   00B6   300          BGTRU   30$                             ;  terminator character
          19 FF43 CF   50   E0   00B8   301          BBS     R0,W^NT$STM_MASK,60$        ; Branch on stream terminator
                   55   02   C0   00BE   302  30$:    ADDL2   #2,R5                       ; Increase record length for CRLF
             00CA C7   55   B1   00C1   303          CMPW    R5,NWA$W_DAPBUFSIZ(R7)      ; Make sure message will fit in buffer
                   03   1B   00C6   304  40$:    BLEQU   50$                             ; Branch if record will fit
                00A7   31   00C8   305          BRW     ERRRSZ                          ; Branch (record is too big)
             83 0A0D 8F   B0   00CB   306  50$:    MOVW    #CRLF,(R3)+                 ; Add CRLF to terminate record
             0120 C7   02   C0   00D0   307          ADDL2   #2,NWA$Q_SAVE_DESC(R7)      ; Update record length in descriptor
                   23   11   00D5   308          BRB     100$                            ; Branch to common code
                        00D7   309
                        00D7   310  ;
                        00D7   311  ; The remote file is in STMLF or STMCR format--add a CR or LF to the user
                        00D7   312  ; record if it does not end in a proper terminator.
                        00D7   313  ;
                        00D7   314
                50   0A   9A   00D7   315  60$:    MOVZBL  #LF,R0                      ; Setup terminator to add
                   03   11   00DA   316          BRB     80$                             ;
                50   0D   9A   00DC   317  70$:    MOVZBL  #CR,R0                      ; Setup terminator to add
                  14 A6   B5   00DF   318  80$:    TSTW    BDB$W_NUMB(R6)              ; Is it a null record?
                   06   13   00E2   319          BEQL    90$                             ; Branch if yes
             FF A3   50   91   00E4   320          CMPB    R0,-1(R3)                   ; Is the proper terminator present?
                   10   13   00E8   321          BEQL    100$                            ; Branch if yes
                   55   D6   00EA   322  90$:    INCL    R5                              ; Increase record length
             00CA C7   55   B1   00EC   323          CMPW    R5,NWA$W_DAPBUFSIZ(R7)      ; Will message fit in buffer?
                   7F   1A   00F1   324          BGTRU   ERRRSZ                          ; Branch if record is too big
                83 50   90   00F3   325          MOVB    R0,(R3)+                    ; Add the terminator
                0120 C7   D6   00F6   326          INCL    NWA$Q_SAVE_DESC(R7)         ; Update record length in descriptor
                        00FA   327
                        00FA   328  ;
                        00FA   329  ; End of record conversion.
                        00FA   330  ;
                        00FA   331
             55    53   D0   00FA   332  100$:   MOVL    R3,R5                       ; Save next byte pointer
                FF00'   30   00FD   333          BSBW    NT$BUILD_TAIL               ; Finish building message
                   15   E1   0100   334          BBC     #DAP$V_DAPCRC,-             ; Branch if partner does not support
                11 28 A7        0102   335                  DAP$Q_SYSCAP(R7),110$       ;  file level CRC checksum
          52    0120 C7   7D   0105   336          MOVQ    NWA$Q_SAVE_DESC(R7),R2      ; Put descriptor of record in <R2,R3>
                0000'CF   0B   010A   337          CRC     W^NT$CRC_TABLE,-            ; Compute CRC (destroying R0-R3)
                  20 A7         010E   338                  DAP$L_CRC_RSLT(R7),-        ;  using result of previous CRC
                63    52        0110   339                  R2,(R3)                     ;  calculation as initial CRC value
             20 A7    50   D0   0112   340          MOVL    R0,DAP$L_CRC_RSLT(R7)       ; Store CRC resultant value
                FEE7'   30   0116   341  110$:   BSBW    NT$TRANSMIT                 ; Send record
                43 50   E9   0119   342          BLBC    R0,CHKSTS                   ; Branch on failure
          0A A6    02   8A   011C   343          BICB2   #<1@BDB$V_DRT>,BDB$B_FLGS(R6) ; Say not dirty anymore
                        0120   344
                        0120   345  ;+
                        0120   346  ; Receive DAP Status message from partner if we are in record transfer mode
                        0120   347  ; and return record file address of the first block accessed.
                        0120   348  ;-
                        0120   349
                        0120   350  PUT_RECV_STS:                                    ;
          61 6A    2D   E0   0120   351          BBS     #IFB$V_SQO,(R10),EXIT       ; Branch if in file transfer mode
          5D 67    24   E1   0124   352          BBC     #DAP$V_GEQ_V56,(R7),EXIT    ; Branch if partner uses DAP before V5.6
                        0128   353  ; ***** $SETBIT #DAP$K_STS_MSG,DAP$L_MSG_MASK(R7); Implied for receive
```

```
                FED5'  30  0128   354          BSBW    NT$RECEIVE              ; Obtain status of put/update request
              31 50    E9  012B   355          BLBC    R0,CHKSTS               ; Branch on failure
               FECF'   30  012E   356          BSBW    NT$RET_RFA              ; Return RFA value to user RAB
               FECC'   30  0131   357          BSBW    NT$RET_RRN              ; Return relative record number in BKT
                               0134   358                                      ;  field (if appropriate)
                 4F    11  0134   359          BRB     EXIT                    ; Branch aid
                               0136   360
                               0136   361 ;+
                               0136   362 ; Access by relative record number or by key value has been requested.
                               0136   363 ; This operation must be performed in DAP record transfer mode.
                               0136   364 ;-
                               0136   365
                               0136   366 PUT_KEY:                            ;
           46 6A    2D  E0     0136   367          BBS     #IFB$V_SQO,(R10),ERRFTM ; Disallow file transfer mode
                50    04  D0     013A   368          MOVL    #DAP$K_CTL_MSG,R0       ; Get message type value
                   FEC0'   30  013D   369          BSBW    NT$BUILD_HEAD           ; Construct message header
                85    04  90     0140   370          MOVB    #DAP$K_PUT_WRITE,(R5)+  ; Store CTLFUNC field
           20   00C6 C7  91     0143   371          CMPB    NWA$B_ORG(R7),#NWA$K_IDX; Branch if IDX organization
                   0C    13  0148   372          BEQL    10$                     ;
                85    0B  90     014A   373          MOVB    #<<DAP$M_RAC>!-         ; Store CTLMENU field
                                 014D   374                    <DAP$M_KEY>!-         ;  (must send relative record number
                                 014D   375                    <DAP$M_ROP>!-         ;  in key field)
                                 014D   376                  0>,(R5)+                ;
                85    01  90     014D   377          MOVB    #DAP$K_KEY_ACC,(R5)+    ; Store RAC field
                   FEAD'   30  0150   378          BSBW    NT$OBTAIN_KEY           ; Store key value
                   FEFE    31  0153   379          BRW     PUT_SEND_CTL_COMMON     ; Join common code
                85    09  90     0156   380 10$:    MOVB    #<<DAP$M_RAC>!-         ; Store CTLMENU field
                                 0159   381                    <DAP$M_ROP>!-         ;  (key string is implicit in data)
                                 0159   382                  0>,(R5)+                ;
                85    01  90     0159   383          MOVB    #DAP$K_KEY_ACC,(R5)+    ; Store RAC field
                   FEF5    31  015C   384          BRW     PUT_SEND_CTL_COMMON     ; Join common code
                               015F   385
                               015F   386 ;+
                               015F   387 ; Common exit code for $PUT and $UPDATE.
                               015F   388 ;-
                               015F   389
           22 67    08  E0     015F   390 CHKSTS: BBS     #NWA$V_UPDATE,(R7),EXIT ; Branch if this is an update operation
                30 A7    91  0163   391          CMPB    DAP$B_TYPE(R7),-        ; Branch if failure was not the result
                   09            0166   392                  #DAP$K_STS_MSG          ;  of Status message returned by FAL
                   1C    12  0167   393          BNEQ    EXIT                    ;
                   01    BB  0169   394          PUSHR   #^M<R0>                 ; Save primary error code
                 FE92'   30  016B   395          BSBW    NT$RESUME_FAL           ; Tell FAL what to do on put error via
                               016E   396                                          ;  interrupt Continue Transfer message
                   01    BA  016E   397          POPR    #^M<R0>                 ; Restore primary error code
                   13    11  0170   398          BRB     EXIT                    ;
                               0172   399 ERRRSZ: RMSERR  RSZ                     ; Invalid record size
                   0C    11  0177   400          BRB     EXIT                    ;
                               0179   401 ERRRAC: RMSERR  RAC                     ; Invalid record access value
                   05    11  017E   402          BRB     EXIT                    ;
                               0180   403 ERRFTM: RMSERR  FTM                     ; File transfer mode error
                               0185   404 EXIT:   $CLRBIT #NWA$V_UPDATE,(R7)     ; Initialize flag for next time thru
                 0C A9   7C    0189   405          CLRQ    IRB$L_IOS(R9)           ; Zero I/O status block
                               018C   406                                          ;  (pertinent status info is already
                               018C   407                                          ;  in R0 and RAB$L_STV)
              00F0 8F    BA  018C   408          POPR    #^M<R4,R5,R6,R7>        ; Restore registers
                         05  0190   409          RSB                             ; Exit with RMS code in R0
```

```
0191    411          .SBTTL  NT$UPDATE - PERFORM NETWORK UPDATE FUNCTION
0191    412
0191    413 ;++
0191    414 ; NT$UPDATE - engages in a DAP dialogue with the remote FAL to update (modify)
0191    415 ;       the specified record of a sequential, relative, or indexed file.
0191    416 ;
0191    417 ; Calling Sequence:
0191    418 ;
0191    419 ;       BSBW    NT$UPDATE
0191    420 ;
0191    421 ; Input Parameters:
0191    422 ;
0191    423 ;       R4      BDB address
0191    424 ;       R8      RAB address
0191    425 ;       R9      IRAB address
0191    426 ;       R10     IFAB address
0191    427 ;       R11     Impure Area address
0191    428 ;
0191    429 ; Implicit Inputs:
0191    430 ;
0191    431 ;       BDB buffer contents
0191    432 ;       BDB$L_ADDR
0191    433 ;       BDB$W_NUMB
0191    434 ;       DAP$V_DAPCRC
0191    435 ;       DAP$V_GEQ_V56
0191    436 ;       DAP$L_CRC_RSLT
0191    437 ;       IFB$L_NWA_PTR
0191    438 ;       IFB$V_SQO
0191    439 ;       NWA$B_RFM
0191    440 ;       NWA$W_DAPBUFSIZ
0191    441 ;       NWA$Q_BLD
0191    442 ;       RAB$B_RAC
0191    443 ;       RAB$L_ROP
0191    444 ;
0191    445 ; Output Parameters:
0191    446 ;
0191    447 ;       R0      Status code (RMS)
0191    448 ;       R1-R7   Destroyed
0191    449 ;       AP      Destroyed
0191    450 ;
0191    451 ; Implicit Outputs:
0191    452 ;
0191    453 ;       BDB$B_FLGS
0191    454 ;       DAP$L_CRC_RSLT
0191    455 ;       IRB$L_IOS and IRB$L_IOS4 zeroed
0191    456 ;       RAB$W_RFA
0191    457 ;
0191    458 ; Completion Codes:
0191    459 ;
0191    460 ;       Standard RMS completion codes
0191    461 ;
0191    462 ; Side Effects:
0191    463 ;
0191    464 ;       None
0191    465 ;
0191    466 ;--
0191    467
```

```
                     0191   468 NT$UPDATE::                             ; Entry point
                     0191   469         $TSTPT  NTUPDATE                ;
   00F0 8F    BB     0197   470         PUSHR   #^M<R4,R5,R6,R7>        ; Save registers
E1 6A   2D    E0     019B   471         BBS     #IFB$V_SQO,(R10),ERRFTM ; Network update function not allowed
                     019F   472                                         ;  if file transfer mode selected
   56   54    D0     019F   473         MOVL    R4,R6                   ; Copy BDB address
57    3C AA    D0     01A2   474         MOVL    IFB$L_NWA_PTR(R10),R7   ; Get address of NWA (and DAP)
                     01A6   475         $SETBIT #NWA$V_UPDATE,(R7)      ; Denote this is an update operation
                     01AA   476
                     01AA   477 ;+
                     01AA   478 ; Build and send DAP Control message to partner.
                     01AA   479 ;-
                     01AA   480
                     01AA   481 UPD_SEND_CTL:                            ;
                     01AA   482         $SETBIT #NWA$V_LAST_MSG,(R7)    ; Declare this last message to block
   50   04    D0     01AE   483         MOVL    #DAP$K_CTL_MSG,R0       ; Get message type value
      FE4C'   30     01B1   484         BSBW    NT$BUILD_HEAD           ; Construct message header
   85   03    90     01B4   485         MOVB    #DAP$K_UPDATE,(R5)+     ; Store CTLFUNC field
   85   09    90     01B7   486         MOVB    #<<DAP$M_RAC>!-         ; Store CTLMENU field
                     01BA   487                   <DAP$M_ROP>!-         ;
                     01BA   488                   0>,(R5)+              ;
   85   00    90     01BA   489         MOVB    #DAP$K_SEQ_ACC,(R5)+    ; Store RAC field--assume SEQ access
01   1E A5    91     01BD   490         CMPB    RAB$B_RAC(R5),#RAB$C_KEY; Keyed access specified?
        04    12     01C1   491         BNEQ    10$                     ; Branch if not
FF A5   01    90     01C3   492         MOVB    #DAP$K_KEY_ACC,-1(R5)   ; Change RAC field
      FE8A    31     01C7   493 10$:     BRW     PUT_SEND_CTL_COMMON     ; Join common code
```

```
                     01CA   495          .SBTTL  NT$DELETE - PERFORM NETWORK DELETE FUNCTION
                     01CA   496
                     01CA   497 ;++
                     01CA   498 ; NT$DELETE - engages in a DAP dialogue with the remote FAL to delete the
                     01CA   499 ;       specified record of a relative or indexed file.
                     01CA   500 ;
                     01CA   501 ; Calling Sequence:
                     01CA   502 ;
                     01CA   503 ;       BSBW    NT$DELETE
                     01CA   504 ;
                     01CA   505 ; Input Parameters:
                     01CA   506 ;
                     01CA   507 ;       R8      RAB address
                     01CA   508 ;       R9      IRAB address
                     01CA   509 ;       R10     IFAB address
                     01CA   510 ;       R11     Impure Area address
                     01CA   511 ;
                     01CA   512 ; Implicit Inputs:
                     01CA   513 ;
                     01CA   514 ;       DAP$V_GEQ_V56
                     01CA   515 ;       IFB$L_NWA_PTR
                     01CA   516 ;       IFB$V_SQO
                     01CA   517 ;       NWA$B_ORG
                     01CA   518 ;
                     01CA   519 ; Output Parameters:
                     01CA   520 ;
                     01CA   521 ;       R0      Status code (RMS)
                     01CA   522 ;       R1-R3   Destroyed
                     01CA   523 ;       AP      Destroyed
                     01CA   524 ;
                     01CA   525 ; Implicit Outputs:
                     01CA   526 ;
                     01CA   527 ;       None
                     01CA   528 ;
                     01CA   529 ; Completion Codes:
                     01CA   530 ;
                     01CA   531 ;       Standard RMS completion codes
                     01CA   532 ;
                     01CA   533 ; Side Effects:
                     01CA   534 ;
                     01CA   535 ;       None
                     01CA   536 ;
                     01CA   537 ;--
                     01CA   538
                     01CA   539 NT$DELETE::                             ; Entry point
                     01CA   540          $TSTPT  NTDELETE               ;
     00F0 8F     BB  01D0   541          PUSHR   #^M<R4,R5,R6,R7>       ; Save registers
  38 6A     2D     E0  01D4   542          BBS     #IFB$V_SQO,(R10),ERRFTM2; Network delete function not allowed
                     01D8   543                                         ;  if file transfer mode selected
     56     54     D0  01D8   544          MOVL    R4,R6                  ; Copy BDB address
  57   3C   AA     D0  01DB   545          MOVL    IFB$L_NWA_PTR(R10),R7  ; Get address of NWA (and DAP)
00   00C6   C7     91  01DF   546          CMPB    NWA$B_ORG(R7),#NWA$K_SEQ; Screen out SEQ file organization
            31     13  01E4   547          BEQL    ERRIOP                 ; Branch on error
                     01E6   548
                     01E6   549 ;+
                     01E6   550 ; Build and send DAP Control message to partner.
                     01E6   551 ;-
```

```
                     01E6   552
                     01E6   553 DEL_SEND_CTL:                                ;
                     01E6   554          $SETBIT #NWA$V_LAST_MSG,(R7)       ; Declare this last message to block
 50    04    D0      01EA   555          MOVL    #DAP$K_CTL_MSG,R0          ; Get message type value
    FE10'    30      01ED   556          BSBW    NT$BUILD_HEAD              ; Construct message header
 85    05    90      01F0   557          MOVB    #DAP$K_DELETE,(R5)+        ; Store CTLFUNC field
 85    01    90      01F3   558          MOVB    #DAP$M_RAC,(R5)+           ; Store CTLMENU field
 85    00    90      01F6   559          MOVB    #DAP$K_SEQ_ACC,(R5)+       ; Store RAC field
    FE04'    30      01F9   560          BSBW    NT$BUILD_TAIL              ; Finish building message
    FE01'    30      01FC   561          BSBW    NT$TRANSMIT                ; Send Control message to FAL
 07 50       E9      01FF   562          BLBC    R0,EXIT2                   ; Branch on failure
                     0202   563
                     0202   564 ;+
                     0202   565 ; Receive DAP Status message from partner.
                     0202   566 ;-
                     0202   567
                     0202   568 DEL_RECV_STS:                                ;
03 67    24    E1    0202   569          BBC     #DAP$V_GEQ_V56,(R7),EXIT2;Branch if partner uses DAP before V5.6
                     0206   570 ; ***** $SETBIT #DAP$K_STS_MSG,DAP$L_MSG_MASK(R7); Implied for receive
    FDF7'    30      0206   571          BSBW    NT$RECEIVE                 ; Obtain status of delete request
                     0209   572
                     0209   573 ;+
                     0209   574 ; Exit code for $DELETE.
                     0209   575 ;-
                     0209   576
 00F0 8F     BA      0209   577 EXIT2:  POPR    #^M<R4,R5,R6,R7>           ; Restore registers
    FDF0'    31      020D   578          BRW     NT$EXRMS                   ; Exit with RMS code in R0
                     0210   579 ERRFTM2:RMSERR  FTM                        ; File transfer mode error
       F2    11      0215   580          BRB     EXIT2                      ;
                     0217   581 ERRIOP: RMSERR  IOP                        ; Illegal I/O operation
       EB    11      021C   582          BRB     EXIT2                      ;
                     021E   583
                     021E   584          .END                               ; End of module
```

```
$$.PSECT_EP                     = 00000000
$$COUNT                         = 00000003
$$RMSTEST                       = 0000001A
$$RMS_PBUGCHK                   = 00000010
$$RMS_TBUGCHK                   = 00000008
$$RMS_UMODE                     = 00000004
BDB$B_FLGS                      = 0C00000A
BDB$L_ADDR                      = 00000018
BDB$V_DRT                       = 00000001
BDB$W_NUMB                      = 00000014
CHKSTS                            0000015F R      01
CR                              = 0000000D
CRLF                            = 00000A0D
CTRLZ                           = 0000001A
DAP$B_BITCNT                      00000035
DAP$B_BLKCNT                      00000056
DAP$B_CTLFUNC                     00000040
DAP$B_DCODE_FID                   00000019
DAP$B_DCODE_MAC                   0000001B
DAP$B_DCODE_MSG                   0000001A
DAP$B_DECVER                      00000047
DAP$B_ECONUM                      00000045
DAP$B_FILESYS                     00000043
DAP$B_FLAGS                       00000031
DAP$B_KRF                         00000047
DAP$B_LEN256                      00000034
DAP$B_LENGTH                      00000033
DAP$B_OSTYPE                      00000042
DAP$B_RAC                         00000046
DAP$B_STREAMID                    00000032
DAP$B_TYPE                        00000030
DAP$B_USRNUM                      00000046
DAP$B_USRVER                      00000048
DAP$B_VERNUM                      00000044
DAP$B_X_FIELD                     00000024
DAP$C_BLN                         000000C0
DAP$K_BLN                         000000C0
DAP$K_CTL_MSG                   = 00000004
DAP$K_DAT_MSG                   = 00000008
DAP$K_DELETE                    = 00000005
DAP$K_KEY_ACC                   = 00000001
DAP$K_PUT_WRITE                 = 00000004
DAP$K_SEQ_ACC                   = 00000000
DAP$K_SEQ_FILE                  = 00000003
DAP$K_STS_MSG                   = 00000009
DAP$K_UPDATE                    = 00000003
DAP$L_CMWA                        00000030
DAP$L_CRC_RSLT                    00000020
DAP$L_DCODE_STS                   00000018
DAP$L_MSG_MASK                    0000001C
DAP$L_RECNUM1                     00000040
DAP$L_ROP                         00000050
DAP$L_SSPWA                       00000080
DAP$L_TEMP                        00000090
DAP$M_BITCNT                    = 00000008
DAP$M_BLKCNT                    = 00000040
DAP$M_KEY                       = 00000002
DAP$M_RAC                       = 00000001
DAP$M_ROP                       = 00000008
DAP$M_SEGMENT                   = 00000040
DAP$M_TMP1$                     = 00000008
DAP$M_TMP2$                     = FFF80000
DAP$Q_DCODE_FLG                   00000000
DAP$Q_FILEDATA                    00000044
DAP$Q_KEY                         00000048
DAP$Q_MSG_BUF1                    00000008
DAP$Q_MSG_BUF2                    00000010
DAP$Q_SYSCAP                      00000028
DAP$Q_SYSPEC                      00000038
DAP$V_DAPCRC                    = 00000015
DAP$V_GEQ_V56                   = 00000024
DAP$W_BUFSIZ                      00000040
DAP$W_CTLMENU                     00000044
DAP$W_DISPLAY2                    00000054
DAP$W_PARTNER                     00000006
DAP$W_VERSION                     00000004
DC1                             = 00000011
DC2                             = 00000012
DC3                             = 00000013
DC4                             = 00000014
DEL_RECV_STS                      00000202 R      01
DEL_SEND_CTL                      000001E6 R      01
DLE                             = 00000010
ERRFTM                            00000180 R      01
ERRFTM2                           00000210 R      01
ERRIOP                            00000217 R      01
ERRRAC                            00000179 R      01
ERRRSZ                            00000172 R      01
ESC                             = 0000001B
EXIT                              00000185 R      01
EXIT2                             00000209 R      01
FAB$C_FIX                       = 00000001
FAB$C_STM                       = 00000004
FAB$C_STMCR                     = 00000006
FAB$C_STMLF                     = 00000005
FAB$C_UDF                       = 00000000
FAB$C_VAR                       = 00000002
FAB$C_VFC                       = 00000003
FF                              = 0000000C
IFB$L_NWA_PTR                   = 0000003C
IFB$V_SQO                       = 0000002D
IRB$L_IOS                       = 0000000C
LF                              = 0000000A
NT$BUILD_HEAD                     ********    X   01
NT$BUILD_TAIL                     ********    X   01
NT$CHK_RAC                        ********    X   01
NT$CRC_TABLE                      ********    X   01
NT$DELETE                         000001CA RG     01
NT$EXRMS                          ********    X   01
NT$MAP_ROP                        ********    X   01
NT$OBTAIN_KEY                     ********    X   01
NT$PUT                            00000004 RG     01
NT$RECEIVE                        ********    X   01
NT$RESUME_FAL                     ********    X   01
```

```
NT$RET_RFA                  ********   X   01
NT$RET_RRN                  ********   X   01
NT$STM_MASK                 00000000 RG    01
NT$TRANSMIT                 ********   X   01
NT$UPDATE                   00000191 RG    01
NWA$B_ALLXABCNT             0000011C
NWA$B_DAP_RAC               000000C9
NWA$B_FILESYS               000000C5
NWA$B_KEYXABCNT             0000011D
NWA$B_NETSTRSIZ             0000016F
NWA$B_NODBUFSIZ             00000168
NWA$B_ORG                   000000C6
NWA$B_OSTYPE                000000C4
NWA$B_RFM                   000000C7
NWA$B_RMS_RAC               000000C8
NWA$C_BLN                   00000800
NWA$K_BLN                   00000800
NWA$K_IDX                 = 00000020
NWA$K_SEQ                 = 00000000
NWA$L_ALLXABADR             00000100
NWA$L_DATXABADR             00000104
NWA$L_DEV                   000000C0
NWA$L_FHCXABADR             00000108
NWA$L_KEYXABADR             0000010C
NWA$L_MSG_MASK              000000D4
NWA$L_PROXABADR             00000110
NWA$L_RDTXABADR             00000114
NWA$L_SAVE_FLGS             00000128
NWA$L_SUMXABADR             00000118
NWA$L_THREAD                000000FC
NWA$L_XLTATTR               00000238
NWA$L_XLTBUFFLG             0000022C
NWA$L_XLTCNT                00000228
NWA$L_XLTMAXINDX            00000234
NWA$L_XLTSIZ                00000230
NWA$Q_ACS                   00000244
NWA$Q_BIGBUF                00000170
NWA$Q_BLD                   000000F0
NWA$Q_FLG                   00000000
NWA$Q_INODE                 0000025C
NWA$Q_IOSB                  00C000D8
NWA$Q_LNODE                 00000160
NWA$Q_LOGNAME               0000023C
NWA$Q_NCB                   00000264
NWA$Q_RCV                   000000E0
NWA$Q_SAVE_DESC             00000120
NWA$Q_XLTBUF1               0000024C
NWA$Q_XLTBUF2               00000254
NWA$Q_XMT                   000000E8
NWA$T_ACSBUF                0000026C
NWA$T_AUXBUF                000005E0
NWA$T_DAP                   00000000
NWA$T_INODEBUF              000004AC
NWA$T_ITM_ATTR              00000200
NWA$T_ITM_END               00000224
NWA$T_ITM_LST               00000200
NWA$T_ITM_MAXINDX           00000218
NWA$T_ITM_STRING            0000020C
NWA$T_NCBBUF                0000052C
NWA$T_NODEBUF               00000169
NWA$T_RCVBUF                000001A0
NWA$T_SCAN                  00000100
NWA$T_TEMP                  00000120
NWA$T_XLTBUF1               000002AC
NWA$T_XLTBUF2               000003AC
NWA$T_XMTBUF                000003C0
NWA$V_FTM_INIT            = 00000019
NWA$V_FTM_RETRV           = 0000001A
NWA$V_FTM_STORE           = 0000001B
NWA$V_LAST_MSG            = 00000000
NWA$V_UPDATE              = 00000008
NWA$W_BUILD                 000000D2
NWA$W_DAPBUFSIZ             000000CA
NWA$W_DIR_OFF               000000CC
NWA$W_DISPLAY               000000D0
NWA$W_FIL_OFF               000000CE
NWA$W_JNLXABJOP             0000011E
PIO$A_TRACE                 ********   X   01
PUT_KEY                     00000136 R     01
PUT_RECV_STS                00000120 R     01
PUT_SEND_CTL_COMMON         00000054 R     01
PUT_SEND_DAT                00000063 R     01
PUT_SEQ                     00000031 R     01
PUT_SEQ_COMMON              00000045 R     01
RAB$B_RAC                 = 0000001E
RAB$C_KEY                 = 00000001
RAB$C_RFA                 = 00000002
RAB$C_SEQ                 = 00000000
RMS$_FTM                  = 000187C4
RMS$_IOP                  = 00018574
RMS$_RAC                  = 00018644
RMS$_RSZ                  = 000186A4
TPT$L_NTDELETE              ********   X   01
TPT$L_NTPUT                 ********   X   01
TPT$L_NTUPDATE              ********   X   01
UPD_SEND_CTL                000001AA R     01
VT                        = 0000000B
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
NF$NETWORK	0000021E (542.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
ABS	00000800 (2048.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.10	00:00:00.84
Command processing	140	00:00:00.54	00:00:03.40
Pass 1	360	00:00:14.10	00:00:38.23
Symbol table sort	0	00:00:01.88	00:00:03.69
Pass 2	112	00:00:02.74	00:00:08.55
Symbol table output	25	00:00:00.17	00:00:00.54
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	678	00:00:19.55	00:00:55.44

The working set limit was 1650 pages.
75704 bytes (148 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1371 non-local and 24 local symbols.
584 source lines were read in Pass 1, producing 15 object records in Pass 2.
29 pages of virtual memory were used to define 28 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[RMS.OBJ]RMS.MLB;1	20
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	24

1582 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:NTOPUT/OBJ=OBJ$:NTOPUT MSRC$:NTOPUT/UPDATE=(ENH$:NTOPUT)+LIB$:RMS/LIB

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NT0SCNXAB
LIS

RM0CACHE
LIS

NT0RENAME
LIS

NT0OPEN
LIS

RM0BUFMGR
LIS

NT0SEARCH
LIS

NT0PUT
LIS

RM0ACCESS
LIS